Diagnosability Analysis of Discrete Event Systems with Autonomous Components

Lina YE and **Philippe DAGUE** [1]

Abstract. Diagnosability is the property of a given partially observable system model to always exhibit unambiguously a failure behavior from its only available observations in finite time after the fault occurrence, which is the basic question that underlies diagnosis taking into account its requirements at design stage. However, for the sake of simplicity, the previous works on diagnosability analysis of discrete event systems (DESs) have the same assumption that any observable event can be globally observed, which is at the price of privacy. In this paper, we first briefly describe cooperative diagnosis architecture for DESs with autonomous components, where any component can only observe its own observable events and thus keeps its internal structure private. And then a new definition of cooperative diagnosability is consequently proposed. At the same time, we present a formal framework for cooperative diagnosability checking, where global consistency of local diagnosability analysis can be achieved by analyzing communication compatibility between local twin plants without any synchronization. The formal algorithm with its discussion is provided as well.

1 INTRODUCTION

Automated fault diagnosis has a significant economic impact on the improvement of performance and reliability of complex discrete event systems (DESs). This problem has received considerable attention from Artificial Intelligence community and Control community. Generally speaking, diagnosis reasoning is to detect possible faults that can explain the observations continuously received by a monitor from a system. However, the accuracy of diagnosis depends on diagnosability of the system. Diagnosability is the property of a given partially observable system model to always exhibit unambiguously a failure behavior from its only available observations in finite time after the fault occurrence, which is the basic question that underlies diagnosis taking into account its requirements at design stage.

The idea of classical and centralized diagnosability analysis methods is to check the existence of indistinguishable behaviors leading to different diagnosis decisions through either deterministic diagnoser ([5]) or twin plant construction ([2] and [8]). However, their knowledge about the system is assumed to be a monolithic model, one automaton representing the entire complex system. This hypothesis is normally unrealistic when dealing with real complex systems due to the combinatorial explosion of the state space. So recently some distributed approaches for this problem have been investigated ([3], [4], [6] and [7]) to avoid building global objects. These distributed approaches have whereas the same assumption that all observable events in any component are globally observed, which means that there is still some global knowledge available and thus at the price of privacy. We propose here a new formal framework for checking diagnosability of DESs with autonomous components, where any component can only observe its own observable events and thus keeps its internal structure private.

There are several objectives of this paper. The first one is the new cooperative diagnosis architecture proposed for DESs with autonomous components. Instead of deciding diagnosis with a sequence of global observations, the idea of cooperative diagnosis is to make diagnosis decision with the cooperation of different components by exchanging local diagnosis decisions with communication information and then by analyzing their communication compatibility. The second one is to define the new definition of cooperative diagnosability for the cooperative diagnosis architecture with the formal theoretical framework for its verification. Specifically, the original diagnosability information is obtained from the local twin plant of the component where the fault may occur and then its propagation to other local twin plants is done by analyzing communication compatibility without any synchronization, which greatly reduces the search state space. The third one consists in the discussion about the efficiency improvement by adopting a reasonable heuristic to choose the next component for further exploitation in the algorithm.

The paper is organized as follows. In the next section, we first model a DES with autonomous components as a set of finite state machines (FSMs) and then briefly describe the cooperative diagnosis architecture. Then Section 3 presents the formal theoretical framework for cooperative diagnosability verification before the formal algorithm is provided in Section 4 with its evaluation discussion. Finally, some related works are referred to in Section 5 before the conclusion.

2 PRELIMINARIES

In this section, we first describe how to model DESs with autonomous components and then give some important concepts before proposing cooperative diagnosis architecture for such systems.

2.1 System model

We consider a distributed DES composed of a set of autonomous components $\{G_1, G_2,..., G_n\}$ that communicate with each other by communication events. Moreover, any component can only observe its own observable events and thus can keep its internal structure private. This kind of system is modeled by a set of FSMs with each one representing the local model of one component.

Definition 1 *(Local model). The local model of the component G_i is a FSM, denoted by $G_i = (Q_i, \Sigma_i, \delta_i, q_i^0)$, where*

[1] University of Paris South, LRI, CNRS / INRIA Saclay, Ile-de-France, France, email: name.surname@lri.fr

- Q_i *is the set of states;*
- Σ_i *is the set of events;*
- $\delta_i \subseteq Q_i \times \Sigma_i \times Q_i$ *is the set of transitions;*
- q_i^0 *is the initial state.*

The set of events Σ_i is partitioned into four subsets: Σ_{i_o}, the set of locally observable events, that can be observed only by its own component G_i; Σ_{i_u}, the set of unobservable normal events; Σ_{i_f}, the set of unobservable fault events, and Σ_{i_c}, the set of unobservable communication events shared by at least one other component, which are the only shared events between components.

For the transition set, it is easy to extend $\delta_i \subseteq Q_i \times \Sigma_i \times Q_i$ to $\delta_i \subseteq Q_i \times \Sigma_i^* \times Q_i$ in the following way: 1) $(q, \epsilon, q) \in \delta_i$, where ϵ is the null event; 2) $(q, se, q1) \in \delta_i$ if $\exists q\prime \in Q_i, (q, s, q\prime) \in \delta_i$ and $(q\prime, e, q1) \in \delta_i$, where $s \in \Sigma_i^*, e \in \Sigma_i$.

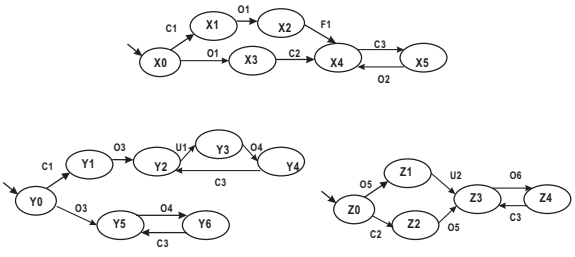

Figure 1: A system with three autonomous components: G_1 (top), G_2 (bottom left) and G_3 (bottom right)

Figure 1 depicts a system with three autonomous components: the events Oi denote locally observable events; the events Fi denote unobservable fault events; the events Ui denote unobservable normal events and the events Ci denote unobservable communication events. Now we define the operation of synchronization between two FSMs.

Definition 2 *(Synchronization). Given two FSMs* $G_1 = (Q_1, \Sigma_1, \delta_1, q_1^0)$ *and* $G_2 = (Q_2, \Sigma_2, \delta_2, q_2^0)$*, their synchronization is* $G_1 \|_{\Sigma_s} G_2 = (Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, \delta_{1\|2}, (q_1^0, q_2^0))$*, where* $\Sigma_s = \Sigma_1 \cap \Sigma_2$ *is the set of shared events and* $\delta_{1\|2}$ *is defined as follows:*

- $((q_1, q_2), \sigma, (q_1', q_2')) \in \delta_{1\|2}$*, if* $\sigma \in \Sigma_s, (q_1, \sigma, q_1') \in \delta_1$ *and* $(q_2, \sigma, q_2') \in \delta_2$*;*
- $((q_1, q_2), \sigma, (q_1', q_2)) \in \delta_{1\|2}$*, if* $\sigma \notin \Sigma_s$ *and* $(q_1, \sigma, q_1') \in \delta_1$*;*
- $((q_1, q_2), \sigma, (q_1, q_2')) \in \delta_{1\|2}$*, if* $\sigma \notin \Sigma_s$ *and* $(q_2, \sigma, q_2') \in \delta_2$*.*

This operation can be extended to a set of FSMs by using its associativity property. In the synchronization, any shared event always occurs simultaneously in all components that define it and the result is a FSM whose state space is the Cartesian product of the state spaces of the components. As a matter of fact, the global model of the entire system is implicitly defined as the synchronized product of all component models based on their shared events, here communication events. However, the global model will not be calculated in our paper considering that the global knowledge of the whole system will be required neither in our cooperative diagnosis architecture nor during our cooperative diagnosability analysis. Details can be found in the next sections. This obviously avoids the combinatorial explosion of the state space. In the following, a subsystem simply denotes a set of components, not the synchronized product of them.

Given a local model G_i, the prefix-closed language $L(G_i)$ describes the local behaviors of the component G_i, where $L(G_i) \subseteq \Sigma_i^*$. Formally, the language $L(G_i)$ is the set of words produced by G_i: $L(G_i) = \{s \in \Sigma_i^* | \exists q \in Q_i, (q_i^0, s, q) \in \delta_i\}$. In the following, we call a word from $L(G_i)$ a local trajectory in G_i and a sequence $q_0\sigma_0 q_1\sigma_1 ...$ a local path in G_i, where $\sigma_0\sigma_1 ...$ is a local trajectory in G_i and for all i, we have $(q_i, \sigma_i, q_{i+1}) \in \delta_i$. Given $s \in L(G_i)$, we denote the post-language of $L(G_i)$ after s by $L(G_i)/s$, formally defined as: $L(G_i)/s = \{t \in \Sigma_i^* : s.t \in L(G_i)\}$. The projection of a trajectory s to locally observable event set Σ_{i_o} of G_i is denoted by $P_i(s)$. All the above notations applied to local model G_i can be applied to the global model G in the same way. In our paper, we assume that both the local behaviors and the local observable behaviors of any component are live, which means that there is no local loop containing only unobservable events.

Definition 3 *(Relative set). Let* G_i *be a component in a system* G*, the* G_i *relative set, denoted by* $\Re_{G_i}$*, is the set of* G_i *relative components, where relative relation over the set of components is defined as follows:*

1. *For a component* G_j*, if* G_j *shares at least one event with* G_i*,* G_j *is a* G_i *relative component,* $G_i \leftrightarrow G_j$*;*
2. *The relative relation is the reflexive and transitive closure of the relation defined by point* 1*.*

Since the relation defined by point 1 of definition 3 is symmetric, with point 2, the relative relation is actually an equivalence relation.

Definition 4 *(Communication compatibility). In a system* G*, two local trajectories* $s \in L(G_i), s\prime \in L(G_j)$*, where* $G_i \neq G_j$*, are communication compatible if they satisfy the following conditions:*

- $\forall \sigma \in s, \sigma \in \Sigma_{i_c}$*, if* $\sigma \in \Sigma_{j_c}$*, then* $\sigma \in s\prime$*;*
- $\forall \sigma \in s\prime, \sigma \in \Sigma_{j_c}$*, if* $\sigma \in \Sigma_{i_c}$*, then* $\sigma \in s$*;*
- $\forall (\sigma, \sigma\prime)$*, where* $\sigma \in s, \sigma\prime \in s, \sigma \in s\prime, \sigma\prime \in s\prime$*, the occurrence order of* $\sigma, \sigma\prime$ *in* s *is the same as that in* $s\prime$*.*

Two local trajectories s of G_i and $s\prime$ of G_j are communication compatible if for any communication event σ in $s(s\prime)$ that is also contained in the communication event set of $G_j(G_i)$, σ occurs also in $s\prime(s)$ and if all common communication events of s and $s\prime$ have the same occurrence order. The communication compatibility of two local trajectories in different components implies that they will not be blocked when these two components are synchronized. This means that communication information exchanging is sufficient to analyze synchronization between components, which will play a central role in our diagnosability analysis.

Lemma 1 *In a system* G*, given two components* G_i *and* G_j*,* $G_i \neq G_j$*, if* $\Sigma_{i_c} \cap \Sigma_{j_c} = \emptyset$*, then* $\forall (s, s\prime), s \in L(G_i), s\prime \in L(G_j)$*,* s *is communication compatible with* $s\prime$*.*

This can be directly proved by definition 4. If there is no shared event between two different components, then any local trajectory in one component is communication compatible with any one in another component.

2.2 Cooperative diagnosis architecture

In a system with autonomous components, to keep the internal structure private, each component can only observe its own locally observable events. When a fault f occurs in one component G_f, since

G_f has no knowledge of internal structures of other components, sometimes it cannot make a diagnosis decision by itself and thus may require the cooperation of some other components that can often help in deciding more precise diagnosis. To be clear, in the following, G_f denotes the component where the considered fault f may occur.

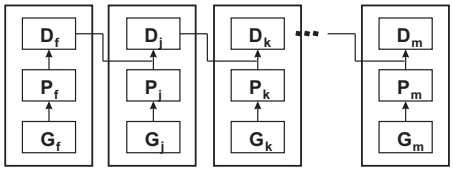

Figure 2: Cooperative diagnosis architecture for systems with autonomous components.

Figure 2 illustrates our cooperative diagnosis architecture. G_f is the component where the considered fault may occur and G_j,G_k ,..., G_m denote other components of the system. Each block P_i denotes the projection of a local trajectory of G_i to the locally observable event set Σ_{i_o}, which is to obtain local observations. D_i is the local diagnosis inference block for the component G_i. Now we briefly describe the idea of diagnosis procedure in this architecture as follows:

1. This procedure begins with the component G_f since it contains the original diagnosis information. Its diagnosis block D_f can infer all possible local trajectories in G_f with their diagnosis from local observable projection P_f. And thus the sequences of local communication events in these corresponding trajectories can be obtained.
2. Regarding all other components, we can deduce all possible local trajectories without diagnosis information with only their local observable projection. If D_f cannot make diagnosis decision by itself, it then informs the next component, suppose G_j, of its inference, i.e., all possible diagnoses with their associated sequence of local communication events in G_f. Note here that we only exchange the communication information between components and thus still keep their internal structure private.
3. All possible local trajectories in G_j deduced from P_j can be synthesized with the inference of D_f by analyzing communication compatibility of possible local trajectories in G_f and those in G_j, where the communication information is sufficient. D_j can thus update possible diagnoses with their associated sequences of local communication events in both G_f and G_j, consistent with P_f and P_j. In other words, D_j keeps communication compatible local trajectories in G_f and G_j with their associated diagnosis and discards the incompatible ones. If D_j still cannot decide diagnosis, then in the same way, D_j informs the next component of its inference. We repeat this step until one component can achieve a final diagnosis decision.

3 THEORETICAL FRAMEWORK

Now we first recall classical diagnosability definition before defining cooperative diagnosability for our cooperative diagnosis architecture. Then, its verification strategy is presented, including the local twin plant construction for the component G_f to obtain the original diagnosability information and the way to propagate it to other local twin plants through communication compatibility analysis.

3.1 Cooperative diagnosability

A fault f is diagnosable in a system G iff its occurrence is determinable when enough long events are observed from the system after the occurrence of f, which is formally defined as follows ([5]), where s^f denotes a trajectory in G ending with f and $P(p)$ denotes the projection of the trajectory p to the observable event set Σ_o of G.

Definition 5 *(Diagnosability). A fault f is diagnosable in a system G iff*

$$\forall s^f \in L(G), \exists k \in N, \forall t = L(G)\backslash s^f, |P(t)| > k \Rightarrow \\ \forall p,\ P(p) = P(s^f.t),\ p \text{ contains } f.$$

The above definition states that for each trajectory s^f in G, for each t that is an extension of s^f in G with sufficiently long observable events, every trajectory p in G that is observation equivalent to $s^f.t$ should contain in it f. Here the system is assumed to have a monolithic model and the observable events are globally observed. The diagnosability checking consists in searching for a pair of trajectories p and $p\prime$ satisfying the following conditions: 1) p contains f and $p\prime$ does not; 2) p has arbitrarily long observations after the occurrence of f; 3) $P(p) = P(p\prime)$. Such a pair is called a **critical pair** [1], which witnesses non-diagnosability.

Unlike the case where a monolithic model exists, our cooperative diagnosis architecture implies that no one has the global knowledge of the whole system and each component is autonomous. Definition 5 can be rephrased to be suitable for systems with autonomous components, which we called cooperative diagnosability. A fault f is cooperatively diagnosable in a system iff for each trajectory s^f ending with the fault f, after any extension t with enough long local observations of all components, we can be sure that f has effectively occurred.

Definition 6 *(Cooperative diagnosability). A fault f is cooperatively diagnosable in a system G, with a set of autonomous components $\{G_1, ...G_n\}$, iff*

$$\forall s^f \in L(G), \exists k \in N, \forall t \in L(G)\backslash s^f, (\forall i \in \{1, ..., n\}, |P_i(t)| > \\ k) \Rightarrow \forall p \in L(G)\ (\forall i \in \{1, ..., n\}, P_i(p) = P_i(s^f.t)) \rightarrow f \in p.$$

In the cooperative diagnosability, we can define a pair of trajectories p and $p\prime$, called **undecidable pair**, as follows, which is similar to critical pair in the classical diagnosability described as above: 1) p contains f and $p\prime$ does not; 2) p has arbitrarily long local observations of all components after the occurrence of f; 3) $\forall i \in \{1, ..., n\}$, $P_i(p) = P_i(p\prime)$. The main difference between a critical pair and an undecidable pair is that for a critical pair, the two trajectories have the same enough long global observations (the same global occurrence order) with only one containing the fault, but the two trajectories of an undecidable pair have the same enough long local observations in each component without considering their global occurrence order. From section 2.2, with such an undecidable pair, there must exist at least one trajectory for which no diagnosis algorithm in our diagnosis architecture can deduce the correct decision and thus f is not cooperatively diagnosable in the system.

Now we are ready to state the following fundamental theorem.

Theorem 1 *A fault f is cooperatively diagnosable in a system G iff there is no undecidable pair in G.*

3.2 Local twin plant

The basic idea of a twin plant, described in [2], is to build a FSM that compares every pair of trajectories to search for the pairs with the same observations, but such that exactly one of them contains a fault, i.e., critical pairs. We now describe how to construct local twin plants of components. We first construct the local diagnoser for a given component, which in turn serves to compute the corresponding local twin plant.

Definition 7 *(Local diagnoser). The local diagnoser of the component G_i is a FSM, denoted by $D_i = (Q_{D_i}, \Sigma_{D_i}, \delta_{D_i}, q^0_{D_i})$ where*

- $Q_{D_i} \subseteq Q_i \times F$, $F \subseteq 2^{\Sigma_{i_f}}$ *is the set of states;*
- $\Sigma_{D_i} = \Sigma_{i_o} \cup \Sigma_{i_c}$ *is the set of events;*
- $\delta_{D_i} \subseteq Q_{D_i} \times \Sigma_{D_i} \times Q_{D_i}$ *is the set of transitions;*
- $q^0_{D_i} = (q^0_i, \emptyset)$ *is the initial state.*

The transitions of δ_{D_i} are those $((q, q_f), e, (q\prime, q_f\prime))$ satisfying the following condition, with (q, q_f) reachable from the initial state $q^0_{D_i}$: there is a transition path $p = (q \xrightarrow{uo_1} q_1 ... \xrightarrow{uo_m} q_m \xrightarrow{e} q\prime)$ in G_i, with $uo_k \in \Sigma_{i_u} \cup \Sigma_{i_f}, \forall k \in \{1, ..., m\}$, $e \in \Sigma_{i_o} \cup \Sigma_{i_c}$ and $q_f\prime = q_f \cup (\{uo_1, ..., uo_m\} \cap \Sigma_{i_f})$.

Then, the local twin plant of a component is obtained by synchronizing its local diagnoser with itself based on the locally observable events, which is to obtain all pairs of local trajectories with the same local observations. The two identical local diagnosers are denoted by D^l_i and D^r_i, left instance and right instance. Since this synchronization is based on the set of locally observable events Σ_{i_o}, the non-synchronized events are distinguished between the two instances by prefixing them with L and R: in D^l_i (D^r_i), each communication event $c \in \Sigma_{i_c}$ from D_i is renamed by $L : c$ ($R : c$) and all their locally observable events unchange their names.

Definition 8 *(Local twin plant). The local twin plant of the component G_i is a FSM, denoted by $T_i = D^l_i \|_{\Sigma_{i_o}} D^r_i$, where D^l_i and D^r_i are the left instance and right instance of the local diagnoser D_i of G_i.*

Each state of a local twin plant is a pair of local diagnoser states that provide two possible diagnoses with the same local observations. Given a twin plant state $((q^l, q^l_f)(q^r, q^r_f))$, if the considered fault $f \in q^l_f \cup q^r_f$ but $f \notin q^l_f \cap q^r_f$, which means that the occurrence of f is not certain in this state, then this twin plant state is called an ambiguous state with respect to the fault f. An ambiguous state cycle is a cycle containing only ambiguous states. In a local twin plant, if a path contains an ambiguous state cycle with at least one locally observable event, then it is called a **local critical path**, which corresponds to a pair of local trajectories with the same local observations but exactly one of them contains the occurrence of the considered fault. Note that local critical paths contain original diagnosability information and can be obtained only in the local twin plant of the component G_f.

Figure 3 presents the local diagnoser and a part of the local twin plant of the component G_1. In the local twin plant, every state is composed of one state label of D^l_i (top) and one state label of D^r_i (bottom). The gray nodes represent ambiguous states with respect to $F1$, which form an ambiguous state cycle. So the part of the local twin plant depicted here is actually a local critical path since it contains an ambiguous state cycle with one locally observable event $O2$.

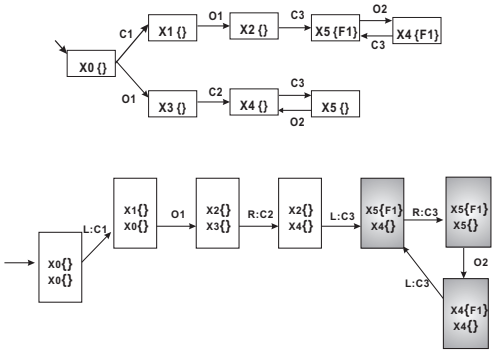

Figure 3: Local diagnoser D_1 (top) and a part of local twin plant T_1 (bottom) of component G_1.

3.3 Diagnosability information propagation

The existence of a local critical path in the local twin plant of the component G_f does not imply that f is not cooperatively diagnosable because its corresponding pair of trajectories in the system is not necessarily an undecidable pair even though they are indistinguishable in G_f. In other words, there may exist another component, suppose G_i, whose cooperation can possibly distinguish this pair when the local observations of its two trajectories in G_i are different. However, since only the component G_f contains the fault information, the projection of any undecidable pair on G_f must correspond to a local critical path in the local twin plant of G_f. Thus the cooperative diagnosability verification consists in checking the existence of local critical paths that correspond to undecidable pairs.

A path ϱ_i in the local twin plant of G_i is communication compatible with a path ϱ_j in the local twin plant of G_j if the corresponding left (right) trajectory of ϱ_i in G_i is communication compatible with the corresponding left (right) trajectory of ϱ_j in G_j. For example, figure 4 presents a part of local twin plants T_2, T_3 of the components G_2, G_3, respectively. The path of T_2 and the path of T_3 depicted here are denoted by ϱ_2 and ϱ_3. The sequence of local communication events in the corresponding left trajectory of ϱ_2 in G_2 is $\{C1, C3^*\}$ and that in the left trajectory of ϱ_3 in G_3 is $\{C3^*\}$. Note that $C1$ is not contained in G_3, then from definition 4, these two trajectories are communication compatible. In the same way, the corresponding right trajectory of ϱ_2 in G_2 and that of ϱ_3 in G_3 are also communication compatible. Thus ϱ_2 is communication compatible with ϱ_3.

Definition 9 *(Local critical path compatibility). In a system G, a local critical path ϱ_f in the local twin plant of G_f is (communication) compatible in a subsystem $G\prime$, where $G_f \subseteq G\prime$, if $\forall G_i \in G\prime \backslash \{G_f\}, \exists \varrho_i$, a path with enough long local observations in the local twin plant of G_i, and this set of paths satisfy the following conditions:*

1. *$\forall G_i \in G\prime \backslash \{G_f\}$, ϱ_i is communication compatible with ϱ_f;*
2. *$\forall (i, j), i \neq j, G_i \in G\prime \backslash \{G_f\}, G_j \in G\prime \backslash \{G_f\}$, ϱ_i is communication compatible with ϱ_j.*

If ϱ_f is compatible in $G\prime$, this set of corresponding local paths in the local twin plant of each component in $G\prime$ that are mutually communication compatible with each other are called (communication) **compatible path set for** ϱ_f in $G\prime$. If a local critical path is compatible in the whole system G, then it is globally compatible. From

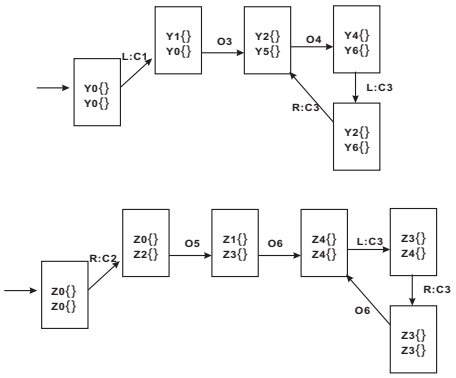

Figure 4: Part of local twin plant T_2 of G_2 (top) and part of local twin plant T_3 of G_3 (bottom).

definition 3 and lemma 1, any path in the local twin plant of any component in $\Re_{G_f}$, G_f relative set, is communication compatible with any path in the local twin plant of any component not in $\Re_{G_f}$. It follows that a local critical path compatible in $\Re_{G_f}$ is globally compatible. From figure 3 and figure 4, the local critical path in the local twin plant T_1, denoted by ϱ_f and presented in figure 3, is communication compatible with both ϱ_2 and ϱ_3. Furthermore, ϱ_2 is communication compatible with ϱ_3. Thus the compatible path set for ϱ_f in the subsystem $\{G_1, G_2, G_3\}$ is $\{\varrho_f, \varrho_2, \varrho_3\}$. Since the system is composed of these three autonomous components, ϱ_f is globally compatible.

Lemma 2 *In a system G, there exists a local critical path that is globally compatible iff there exists an undecidable pair.*

Proof :
($\Rightarrow$) Suppose there exists a local critical path ϱ_f that is globally compatible, but there does not exist an undecidable pair. Since ϱ_f is globally compatible, from definition 9, there must exist a compatible path set for ϱ_f in the whole system. Due to their mutual communication compatibility, this compatible path set, including ϱ_f, correspond to a pair of trajectories in the whole system such that they have the same enough long local observations for all components but exactly one of them contains the fault f, which is actually an undecidable pair (see section 3.1). This follows that there exists an undecidable pair and thus contradicts the assumption.
($\Leftarrow$) Now suppose that there exists an undecidable pair, denoted by p and $p\prime$, but there does not exist a local critical path being globally compatible. The pair p and $p\prime$ being an undecidable pair first implies that they correspond to a local critical path in the local twin plant of G_f, denoted by ϱ_f, and then implies that $\forall i \in \{1, ...n\}$, we have $P_i(p) = P_i(p\prime)$, which forms a path in the local twin plant of each component. Furthermore, since p and $p\prime$ are trajectories in the whole system, their corresponding paths in all local twin plants, including ϱ_f, must be mutually communication compatible with each other and thus constitute a compatible path set for ϱ_f in the whole system. So from definition 9, ϱ_f is globally compatible, which contradicts the assumption.

Lemma 2 with its proof implies the equality between a local critical path that is globally compatible and an undecidable pair. Then from theorem 1 and lemma 2, the major result of this paper can be obtained as follows, which is the theoretical basis of our cooperative diagnosability algorithm for systems with autonomous components.

Theorem 2 *A fault f is cooperatively diagnosable in system G iff there is no local critical path that is globally compatible.*

4 ALGORITHM

Algorithm 1 presents the procedure of verifying cooperative diagnosability through checking the existence of local critical paths being globally compatible. As showed in the pseudo-code for this verification procedure, algorithm 1 performs as follows. Given the input as component models, the component G_f where the considered fault f may occur, we initialize the parameters as empty, i.e., current subsystem $\widehat{G_s}$, set of compatible path sets for local critical paths in current subsystem $\widehat{\varphi}$ and set of connected components under consideration $\widehat{G_c}$. The algorithm begins with the construction of local twin plant of G_f, current subsystem being now G_f and $\widehat{\varphi}$ updated by assigning the set of local critical paths in T_f (line 3-5). After this, if $\widehat{\varphi}$ is not empty (line 6), which implies the existence of compatible path sets for local critical paths in current subsystem and thus the existence of local critical paths compatible in current subsystem, the algorithm repeatedly performs the following steps:

1. If there exists at least one component directly connected with current subsystem, i.e., sharing at least one event, but not involved in current subsystem, the local critical paths compatible in current subsystem need to be further checked for their communication compatibility in an extended subsystem (line 7,8)
2. After selecting a directly connected component G_i, we construct its local twin plant T_i and extend the subsystem by adding G_i (line 9-11)
3. $\widehat{\varphi}$ is now ready to be updated for current extended subsystem. More precisely, before updating, each element in $\widehat{\varphi}$ represents a compatible path set for a local critical path in the precedent subsystem. To update $\widehat{\varphi}$, for each element in it, we check each path of T_i, if one element can become a compatible path set for a local critical path in current extended subsystem by adding one path of T_i, then we update this element by adding this path. Note that each element is updated by adding only one path of T_i but can be updated for several times if there are not only one such path in T_i. In other words, one element in non-updated $\widehat{\varphi}$ can be updated to several elements in updated $\widehat{\varphi}$. Otherwise, if there is no such path in T_i, we remove this element from $\widehat{\varphi}$. In this way, each element of updated $\widehat{\varphi}$ is a compatible path set for a local critical path that is compatible in current extended subsystem (line 12).
4. If $\widehat{\varphi}$ is not empty and there is no component directly connected with current subsystem, then there exists at least one local critical path being globally compatible. In this case, $\widehat{\varphi}$ is returned to provide the corresponding compatible path sets that contain the important information about why f is not cooperatively diagnosable (line 14).

Another reason that can stop the algorithm is the non-existence of local critical path being globally compatible, which is the case of $\widehat{\varphi}$ being empty (line 17). Empty $\widehat{\varphi}$ implies that there is no compatible path set for a local critical path in current subsystem and thus no local critical path being globally compatible, which verifies cooperative diagnosability of the system.

If f is cooperatively diagnosable in the system, we can improve the algorithm efficiency by searching for a subset of $\Re_{G_f}$ that is sufficient to verify cooperative diagnosability through an appropriate component selection strategy. Let Σ_{S_c} be the set of communication events in current subsystem. To choose next component for further compatibility checking, we prefer to select the one, suppose

Algorithm 1 Cooperative Diagnosability Checking Algorithm

1: INPUT: the system model $G = (G_1, ..., G_n)$; G_f, the component where the fault f may occur
2: Initializations: $\widehat{G_s} \leftarrow \emptyset$ (current subsystem considered); $\widehat{\wp} \leftarrow \emptyset$ (set of sets of paths, each element represents a compatible path set for a local critical path in current subsystem); $\widehat{G_c} \leftarrow \emptyset$ (set of components directly connected with current subsystem)
3: $T_f \leftarrow ConstructLTP(G_f)$
4: $\widehat{G_s} \leftarrow \{G_f\}$
5: $\widehat{\wp} \leftarrow Update(\widehat{\wp}, T_f)$
6: **while** $\widehat{\wp} \neq \emptyset$ **do**
7: $\quad \widehat{G_c} \leftarrow CollectDCC(G, \widehat{G_s})$
8: $\quad$ **if** $\widehat{G_c} \neq \emptyset$ **then**
9: $\quad\quad G_i \leftarrow SelectDCC(\widehat{G_c})$
10: $\quad\quad T_i \leftarrow ConstructLTP(G_i)$
11: $\quad\quad \widehat{G_s} \leftarrow ADD(\widehat{G_s}, G_i)$
12: $\quad\quad \widehat{\wp} \leftarrow Update(\widehat{\wp}, T_i)$
13: $\quad$ **else**
14: $\quad\quad$ return $\widehat{\wp}$
15: $\quad$ **end if**
16: **end while**
17: return "f is cooperatively diagnosable in G"

G_i, such that $|\Sigma_{S_c} \cap \Sigma_{i_c}|$, the number of communication events in G_i contained also in the current subsystem, is maximum comparing to other components to be selected. This is a reasonable heuristic because more communication events of the selected component are involved in current subsystem, more likely the compatible path sets for local critical paths in current subsystem will be removed during compatibility checking for the extended subsystem. In this way, the involved components of the algorithm are as few as possible with high probability.

5 RELATED WORKS

In [5], the authors introduced the first definition of diagnosability for DESs and proposed a necessary and sufficient condition for testing it by constructing a deterministic diagnoser for the entire system. The main drawback is its exponential space complexity in the number of system states. And then the authors of [2] and of [8] proposed new algorithms with polynomial complexity in the number of system states, which introduced the classical twin plant method.

However, all above approaches assume the existence of a monolithic model of the entire system, which is normally unrealistic when dealing with real complex systems. This is why recently distributed approaches to solve diagnosability problem are investigated. The diagnosability problem of a system in a distributed way is first introduced in [3] and is solved by synchronizing local twin plants until a global critical path is detected. In the worst case, the global twin plant still cannot be avoided. Then in [7], the proposed approach first decides nondiagnosable states in each local twin plant by propagating diagnosability information. This is done by synchronizing relative local twin plants based on their connectivity with the local twin plant of the component where the fault may occur. Then reduced local twin plants are computed that only contain the parts relevant to solve the diagnosability problem. And thus even in the worst case, the concerned part is a subpart of the global twin plant. Then the authors of [6] present a scalable jointree algorithm to decide diagnosability, where diagnosability information propagation as well as the consistency checking between local twin plants are both done through computing and passing messages on a jointree. Specifically, the global consistency of each local twin plant is checked by synchronizing itself with the corresponding computed message, which is a FSM representing the behavior constraints imposed by other local twin plants. Then diagnosability can be decided on these globally consistent local twin plants.

All these distributed approaches are based on the assumption that some global knowledge is available, i.e., all observable events can be globally observed, which is at the price of privacy. Moreover, with this assumption, these approaches, in the worst case, have exponential complexity in the number of components since they unavoidably synchronize some part of local twin plants to decide diagnosability. To the best of our knowledge, this paper is the first work to propose and solve cooperative diagnosability for DESs with autonomous components, where the privacy of each component is kept as much as possible. Furthermore, the cooperative diagnosability can be determined by only analyzing the communication compatibility between local twin plants without synchronizing any part of them. So even in the worst case, we can avoid exponential complexity in the number of components. With suitable heuristics, the algorithm can further reduce space complexity.

6 CONCLUSION

In this paper, we first describe the cooperative diagnosis architecture for DESs with autonomous components and propose a new definition of cooperative diagnosability. Then a formal framework for cooperative diagnosability checking is put forward. The idea is to search for local critical paths being globally compatible by analyzing communication compatibility between local twin plants, whose existence verifies non-diagnosability. At the same time, we discuss how to improve our algorithm efficiency by adopting a reasonable heuristic and thus with high probability, the involved components are as few as possible. The main perspective to extend this work is the further investigation of how to use our approach to improve the system diagnosability level when it is not cooperatively diagnosable considering that our algorithm returns the set of compatible path sets, which contain the information about why the system is not cooperatively diagnosable.

REFERENCES

[1] A. Cimatti, C. Pecheur, and R. Cavada, 'Formal verification of diagnosability via symbolic model checking', *18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, 363–369, (2003).

[2] S. Jiang, Z. Huang, V. Chandra, and R. Kumar, 'A polynomial time algorithm for diagnosability of discrete event systems', *IEEE Transactions on Automatic Control*, 46(8):1318–1321, (2001).

[3] Y. Pencolé, 'Diagnosability analysis of distributed discrete event systems', *Proceedings of European Conference on Articifial Intelligent ECAI'04*, 43–47, (2004).

[4] Y. Pencolé, 'Assistance for the design of a diagnosable component-based system', *17th IEEE International Conference on Tools with Artificial Intelligence ICTAI'05*, 549–556, (November 2005).

[5] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, 'Diagnosability of discrete event system', *IEEE Transactions on Automatic Control*, 40(9):1555–1575, (1995).

[6] A. Schumann and J. Huang, 'A scalable jointree algorithm for diagnosability', *23rd American National Conference on Artificial Intelligence (AAAI-08)*, (July 2008).

[7] A. Schumann and Y. Pencolé, 'Scalable diagnosability checking of event-driven systems', *20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, 575–580, (2007).

[8] T. Yoo and S. Lafortune, 'Polynomial-time verification of diagnosability of partially observed discrete-event systems', *IEEE Transactions on Automatic Control*, 47(9):1491–1495, (2002).